

时间敏感网络端处理逻辑（TSE） 设计方案 （版本 1.0）

OpenTSN

OpenTSN 开源项目组

2021 年 4 月

版本历史

[illegible]

目录

1、概述.....	4
2、总体设计.....	4
2.1. 总体架构.....	4
2.1.1 主机接收处理逻辑.....	5
2.1.2 主机发送处理逻辑.....	7
2.1.3 网络输入处理逻辑.....	8
2.1.4 网络输出处理逻辑.....	9
2.1.5 内部处理逻辑.....	10
2.2 帧的处理流程.....	11
2.2.1 主机口进网络口出的帧处理流程.....	11
2.2.2 网络口进主机口出的帧处理流程.....	12
附录 1：数据格式定义.....	13
附录 2：内部寄存器定义.....	15
附录 3：NMAC 报文格式.....	21
附录 4：command/command_ack 命令格式.....	24

表 2-1 TSE 总体架构顶层信号定义

信号	位宽	含义
pkt_134（报文集中缓存接收或发送的信号）	134	报文体数据，具体格式参考附录 A
pkt_9（非报文集中缓存模块的信号）	9	报文体数据，具体格式参考附录 A
descriptor	46	报文描述符数据，用于标识报文数据
pkt_bufid	9	报文数据在报文缓存区中缓存的 ID 号
pkt_type	3	报文类型数据
flow_id	14	报文数据的流 ID，用于标识流
lookup_en	1	查表使能
outport	9	输出端口号（bitmap）
inport	4	输入端口 号
queue_id	3	queue_id 号，用于标识队列缓存的 ID 号
multicast_count	4	组播报文的输出端口数量
time_offset	49	1588 同步从时钟架构需要补偿的值
command	204	读、写命令
command_ack	204	读命令响应

下面将整个架构划分为五大模块进行介绍，分别对应顶层中的主机接收处理逻辑、主机发送处理逻辑、网络输入处理逻辑、网络输出处理逻辑、以及内部处理逻辑。

2.1.1 主机接收处理逻辑

主机接收处理逻辑的内部组成框图如图 2-2。

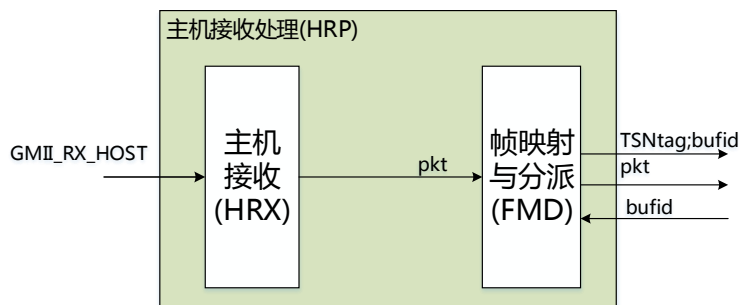


图 2-2 主机接收处理(HRP)模块内部组成框图

HRX(Host RX)网络接收模块：主要功能是接收网络接口发送的报文，完成报文从外部时钟域到架构内部时钟域的切换，以及完成架构接收时间同步报文的时间信息记录，并在 TSNTag 中进行记录。模块内部维护一个寄存器，根据此寄存器的值判断是否接收并处理数据。

FMD(Pkt Map and Dispatch)报文映射与分派模块：主要功能是构造报文描述符（报文描述符的格式如下表 2-2）、对 BE 流和 RC 流进行流量监管以及区分报文类型并分派到不同的目的模块。报文描述符的构造，是根据报文集中缓存模块分配给本接口的 pkt_bufid 以及报文映射后的信息构造一个能够标识报文的描述符数据。流量监管，是根据报文类型进行流量监管，当 bufid 剩余的数量少于 RC 流的阈值时，将 RC 流和 BE 流进行丢弃；当 bufid 剩余的数量少于 BE 流的阈值时，将 BE 流进行丢弃。报文分派，需要先区分报文的类型是 ST、BE 和 RC 还是 NMAC 协议（NMAC 报文格式定义参考附录四），再将不同类型的报文描述符分派到不同的目的模块，并将报文数据也分派到不同目的模块。另外，本模块还需将主机下发报文的 metadata 丢弃。

表 2-2 描述符格式定义

内容	位宽	位置	含义
inject_addr/ submit_addr	5	[45:41]	ST 流注入/提交时缓存的地址。
reserve	1	[40]	保留
inport	4	[39:36]	报文的输入端口，用于最终构造 metadata。
pkt_type	3	[35:33]	报文类型，用于入队控制时区分报文类型，选择队列。

内容	位宽	位置	含义
flow ID/IMAC	14	[32:19]	流 ID，用于 FLT 模块查表时的地址索引。
lookup_en	1	[18]	查表使能，用于 FLT 判断该报文是否需要查表操作。
outport	9	[17:9]	输出端口号，用于 FLT 模块区分输出端口。
pkt_bufid	9	[8:0]	报文在缓存区中缓存的 ID 号，用于标识每个报文。

2.1.2 主机发送处理逻辑

主机发送处理逻辑的内部组成框图如图 2-3。

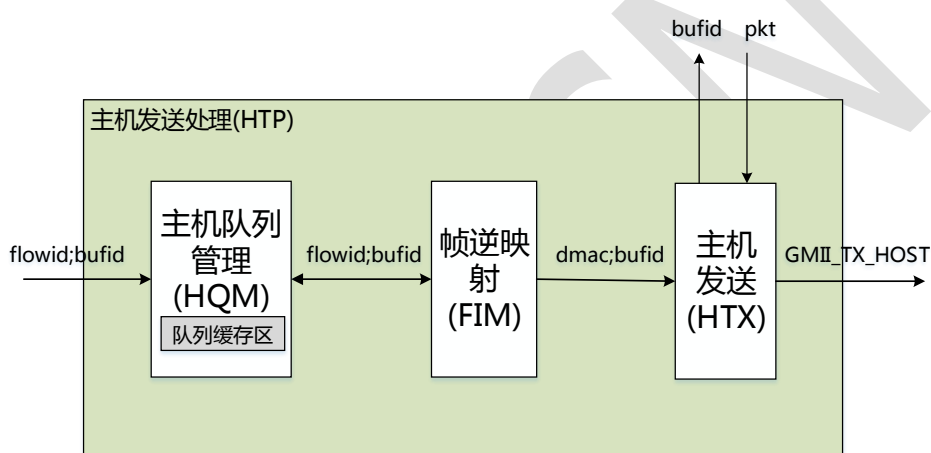


图 2-3 主机发送处理(HTP)模块内部组成框图

FIM(frame inverse mapping)帧逆映射模块：主要功能是将接收到的报文查找逆映射表进行逆映射。

HQM (Host Queue Manage)主机队列管理模块：主要功能是对发往主机的非 ST 流的 pkt_bufid 进行缓存管理，等待主机输出调度模块的调度信号进行 pkt_bufid 的调度输出。

HTX(Host TX)主机发送模块：主要功能是从报文集中缓存区中读取报文并释放 pkt_bufid、将数据报文由主机接口传输给外部主机。读取报文时，需要先将 pkt_bufid 映射成报文读取地址，并根据此地址往报文集中缓存模块进行报文数据的读取，同时需要将此 pkt_bufid

释放给报文集中缓存模块以便后续进入架构的报文使用。当 CSM 需要发送上报报文时，需要构造帧前导符、帧开始符以及 8B 的 metadata 后直接将 CSM 模块发送的上报报文从主机口进行转发。

2.1.3 网络输入处理逻辑

网络输入处理逻辑的内部组成框图如图 2-4。

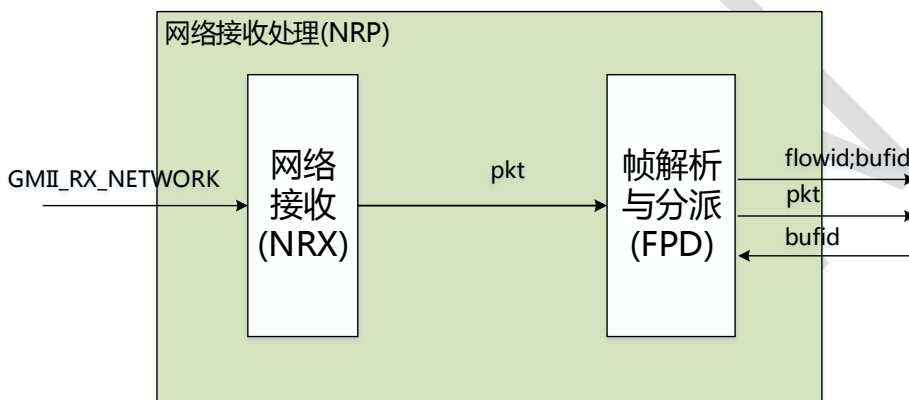


图 2-4 网络输入处理(NRP)模块内部组成框图

NRX(Network RX)网络接收模块：主要功能是接收网络接口发送的报文，完成报文从外部时钟域到架构内部时钟域的切换，以及完成架构接收时间同步报文的时间信息记录，并在 TSNTag 中进行记录。模块内部维护一个寄存器，根据此寄存器的值判断是否接收并处理数据。

FPA(Frame PARse)帧解析模块：主要功能是提取帧的 DMAC (TSNTag) 中的信息，并构造报文描述符数据。还需根据报文的以太网类型字段判断此报文是否经过分类映射，若未经过分类映射则在构造描述符时默认送往主机。在往输入缓存接口模块发送数据之前，需要将一拍 8bit 的数据转换成一拍 128bit 的数据。

2.1.4 网络输出处理逻辑

网络输出处理逻辑的内部组成框图如图 2-5。

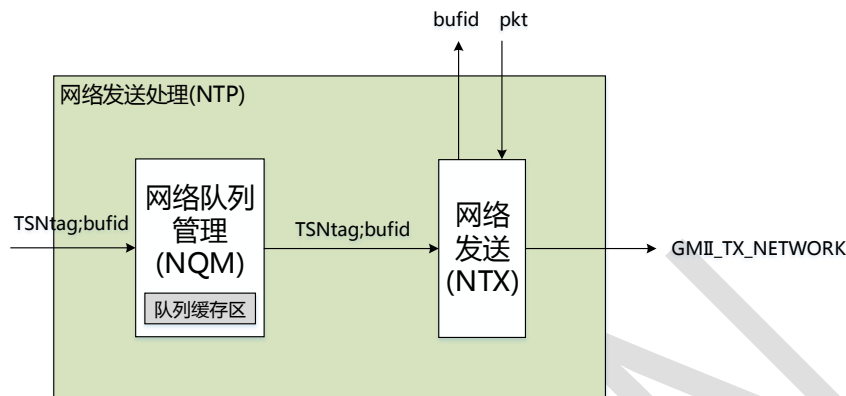


图 2-5 网络输出处理(NTP)模块内部组成框图

NIQ(Network Input Queue)网络入队控制模块：主要功能是将 `pkt_bufid` 写入到网络队列管理模块中进行缓存。本模块需要根据接收到的报文类型信息、队列门控模块发送的门控信息进行 `queue_id` 的映射,并将 `pkt_bufid` 与 `queue_id` 发送给网络队列管理模块进行缓存。同时将 `pkt_bufid` 与 `queue_id` 发送给网络输出调度模块,以便对队列首地址进行管理。本模块还需要根据写入队列的信息与网络输出调度模块传输的调度队列的信号来对队列的状态进行管理,主要是对所有队列中目前所写入的 `bufid` 数量进行管理。

NTX(Network TX)网络发送模块：主要功能是从报文缓存区中读取报文并释放 `pkt_bufid`、计算时间同步报文的透明时钟、完成报文从内部处理时钟域到外部 **PHY** 架构的时钟域的切换、将数据报文构造帧前导符和帧开始符后由网络接口传输。读取报文时,需要先将 `pkt_bufid` 映射成地址,并根据此地址往报文集中缓存模块进行报文数据的提取,同时需要将此 `pkt_bufid` 归还给报文集中缓存模块以便后

续进入架构的报文使用。

2.1.5 内部处理逻辑

PCB(Pkt Centralize Bufm_memory)报文集中缓存模块：主要功能是对架构需要转发的所有报文进行集中缓存（报文缓存区的格式如下表 2-3）、为每个缓存的报文分配一个 pkt_bufid 进行标识、对所有空闲的 pkt_bufid 进行缓存（空闲地址缓存区的格式如下表 2-4。为所有 pkt_bufid 增加寄存器对输出端口数量进行计数，当 pkt_bufid 使用完进行释放的时候需要检测该 pkt_bufid 对应的计数值，只有当计数值为 0 才能进行释放，当不为 0 意味着该 pkt_bufid 存着一个组播报文并且该报文还未从所有需要输出的端口输出。

报文缓存区将 64KB 的空间划分成 512 个报文缓存块，每个报文缓存块能缓存一个长度最多为 128B 的报文。

表 2-3 报文缓存区数据格式

地址[8:0]	内容[133:0]
0-127	第 1 个报文缓存块
128-255	第 2 个报文缓存块
...	..
65408-65535	第 512 个报文缓存块

表 2-4 空闲地址缓存区数据格式

名称	含义	备注
pkt_bufid[8:0]	当前“报文缓存区”中空闲的报文缓存块 ID 号。	使用 RAM 进行实现，深度为 512。

CPA(Command PARse)命令解析模块：负责将接收到 HCP 的 command 命令进行解析，（命令数据格式见表 2-5）来实现对本地寄存器、映射表、逆映射表的配置。周期性的将各个模块的实时状态封装成 command_ack 命令上报给 HCP 模块。

表 2-5 command/command_ack 命令格式

位置	位宽 bit	名称	说明
[203:180]	8	node_id	该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。
[179:172]	8	dest_module_id	该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID
[171:168]	4	type	4' b0001:寄存器或表项的写命令; 4' b0010:寄存器或表项的读命令; 4' b0110:寄存器或表项的读响应。
[167:152]	16	addr	寄存器或表项的读/写地址
[151:0]	152	data	寄存器或表项的读/写数据;其中五元组映射表的表项位宽最大,为 152bit

2.2 帧的处理流程

架构中包含 1 个主机口和 1 个网络口,主机口与外部设备(需接入 TSN 网络中)连接,网络口与 HCP 或 TSS 连接。详细处理流程介绍如下。

2.2.1 主机口进网络口出的帧处理流程

主机口输入的帧先在主机接收模块(HRX)进行跨时钟域处理,并在每拍数据中增加 1bit 的头尾标识位,帧映射与分派模块(FMD)接收到报文后,将报文集中缓存模块(PCB)分配的 bufid 转化为帧写入缓存区的基地址,将报文写到报文集中缓存的缓存区中;同时从报文数据(以太网类型为 0x0800)中提取五元组,用五元组去查找映射表,获得查表结果;以太网类型不是 0x0800 的第一个分片不查

映射表，直接将其 DMAC 作为查表结果；查表后将查表结果 TSntag 和 bufid 传给网络队列管理模块（NQM）进行缓存；网络发送模块接收到 TSntag 和 bufid 后，将 bufid 转化为帧读取的基地址，从报文集中缓存模块（PCB）的缓存区中读出帧并从 GMII 接口输出，在从缓存区中读出帧后，将 bufid 进行释放。

2.2.2 网络口进主机口出的帧处理流程

网络口输入的帧先在网络接收模块（NRX）进行跨时钟域处理，并在每拍数据中增加 1bit 的头尾标识位，帧解析与分派模块（FPD）接收到帧后将报文集中缓存模块（PCB）分配的 bufid 转化为帧写入缓存区的基地址，将报文分组写到报文集中缓存的缓存区中；同时从帧中提取描述符(flowid, bufid)并传输给主机队列管理模块(HQM)，报文分组的描述符缓存到主机队列管理模块（HQM）的队列中；主机队列管理模块将报文分组的 flowid 和 bufid 输出给帧逆映射模块（FIM），在帧逆映射模块（FIM）用 flowid 查找逆映射表，将查表结果 DMAC 和 bufid 输出给主机发送模块(HTX)，在主机发送模块(HTX)将接收到的 bufid 转化为帧读取的基地址，从报文集中缓存模块(PCB)的缓存区中读出报文，从 GMII 接口输出，同时用 DMAC 替换第一个分片中的 TSntag，将以太网类型由 0x1800 改为 0x0800，在从缓存区中读出分片后，将 bufid 进行释放。

附录 1：数据格式定义

● TSNTag 格式

在流量发送端的网卡内部需要根据报文七元组(目的 mac、type、IP 五元组)对时间敏感、带宽预约、尽力转发流量进行分类映射。将分类映射的结果与原报文的 DMAC 字段进行替换，以此进行 TSN 网络的交换，直到接收端的网卡内部进行 DMAC 还原。被替换的 DMAC 字段被定义成 TSNTag。

表附 1-1 分类映射关键字 Key

位宽	名称	描述
48	DMAC	报文目的 MAC
16	ETHTYPE	报文以太网类型
8	protocol	报文协议类型
32	Sip	报文源 ip
32	Dip	报文目的 ip
16	Sport	报文源端口
16	Dport	报文目的端口

因同步报文的 TSNTag 中“seq_id”、“frag_id”、“inject_addr”、“submit_addr”信息是无用的，因此可以将时间同步报文的这些字段用来存放架构的接收时间戳信息。而其他非时间同步报文的架构接收时间戳信息是无用的，因此可以延用这些字段的信息。

表附 1-2 时间同步报文的 TSNTag

位宽	名称	位置	描述
3	Flow type	[47:45]	流类型。 100：同步报文（其他报文的格式如下表）
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。

位宽	名称	位置	描述
12	Reserve	[30:19]	保留
19	Rx_timestamps	[18:0]	架构接收到时间同步报文的本地时间信息，用于架构发送报文时计算透明时钟。

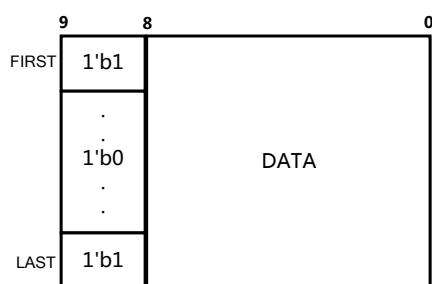
表附 1-3 非时间同步报文的 TSNTag

位宽	名称	位置	描述
3	Flow type	[47:45]	流类型。000:ST 分组 001:ST 分组 010: ST 分组 011: RC 分组 101: NMAC 分组 110: BE 分组 111: BE 分组
14	Flow id/IMAC	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。
16	Seq id	[30:15]	用于标识每条流中报文的序列号
1	Frag flag	[14]	用于标识分片后的尾。0: 分片后的中间报文 1: 尾拍
4	Frag ID	[13:10]	用于表示当前分片报文在原报文中的分片序列号
5	inject addr	[9:5]	ST 流在源端等待发送调度时缓存地址
5	submit addr	[4:0]	ST 流在终端等待接收调度时缓存地址

● 内部传输的 pkt 数据格式

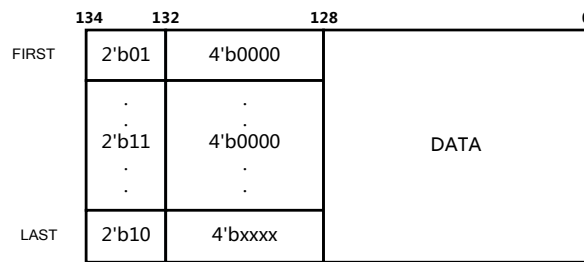
在模块之间传输的有两种格式，一种位宽为 9bit；另一种位宽为 134bit。具体区别如下：

pkt_data 位宽为 9 位，包含 1bit 头尾标志、8bit 报文数据。头尾标志： 1 表示报文头/尾数据；0 标识报文体中间数据。具体如图附 1-1 所示：



图附 1-1 pkt_data 数据格式 1

pkt_data 位宽为 134 位, 包含 2bit 头尾标志、4bit 无效字节、128bit 报文数据。头尾标志: 01 表示报文头; 11 表示报文体中间数据; 10 表示报文尾。无效字节用于标识报文尾中无效的字节数。具体如图附 1-2 所示:



图附 1-2 pkt_data 数据格式 2

附录 2: 内部寄存器定义

架构内部可配置地址空间主要有两部分, 包括: MDID 模块号和真实地址空间, 其中 MDID 模块号主要用来区分不同模块, 而后 20 位为各个模块使用的地址空间。地址的第 19bit 位用于区别地址类型, 控制/表项寄存器可读可写, 调试和版本寄存器只读, 每个模块的地址空间为 1024k, 其中可读可写和只读寄存器各有 512k。具体地址含义如下。

表附 2-1 地址格式

ADDR[26:0]		
MDID[26:20]	ADDR[19]	ADDR[18:0]
MDID : 0-127	0	该模块的控制寄存器, 表项等, 可读 可写
	1	只读

每个处理模块的 MDID 号分配如下:、

表附 2-2 模块中的 MDID 和地址

处理模块	CSM	TIS	TSS	QGC	GTS	FLT
MDID	0x0	0x1	0x2	0x3-0xa	0xb	0xc
地址	0x0-0xfffff	0x100000-0x1fffff	0x200000-0x2ffffff	0x300000-0xafffff	0xb00000-0xbfffff	0xc00000-0xcfffff

● CSM 模块

地址范围为 Addr 0x0-0xffff。

表附 2-3 CSM 模块寄存器

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
0x0	offset_l			
0x1	offset_h			
0x2	time_slot			
0x3	cfg_finish			
0x4	port_type			
0x5	qbv_or_ach			
0x6	report_type			
0x7	report_en			
0x8	inject_slot_period			
0x9	submit_slot_period			
0xa	report_period			
0xb	offset_period			
0xc	rc_regulation_value			
0xd	be_regulation_value			
0xe	unmap_regulation_value			
0xf ~ 0xfffff	reserve			

表附 2-3 寄存器的具体含义

name	bit	R/W	description	default
offset_l	31:1 7	R/W	代表时间偏移的高位值的低 15 位，表示毫秒	0

	16:0	R/W	时间偏移的低位，表示拍数	0
offset_h	31:17	R/W	保留位	0
	16	R/W	代表时间偏移的正负值，1 代表正值，如果为 0，则代表负值	0
	15:0	R/W	代表时间偏移的高位值的高 16 位，表示毫秒	0
time_slot	31:11	R/W	保留	
	10:0	R/W	时间槽大小	0
cfg_finish	31:1	R/W	保留	0
	0	R/W	配置完成寄存器， 0 代表架构正在初始化，不接收任何报文， 1 代表初始化完成，可以接收 NMAC 配置报文 2 代表配置完成，可以接收除 ST 报文的任何报文 3 代表可以接收任何报文	0
port_type	31:8	R/W	保留	0
	7:0	R/W	网络端口类型寄存器，架构共有 8 个网络端口，寄存器的 0-7 位分别代表 0-7 端口的类型，1 代表非合作类型，处理标准以太网类型的报文，0 代表合作类型，处理 TSN 报文	0
qbv_or_ach	31:2	R/W	保留	0
	1:0	R/W	调度模式选择信号，网络输出逻辑中的调度机制是 QBV 模式还是 QCH 模式 0 代表 QBV 模式；1 代表 QCH 模式	0
report_type	31:16	R/W	保留	0
	15:0	R/W	上报类型，具体参考附录 D	0
report_en	31:1	R/W	保留	0
	0	R/W	上报使能信号，配置与状态管理模块是否进行周期性上报 0 代表不上报；1 代表上报	0
inject_slot_period	31:12	R/W	保留	0

	10:0	R/W	注入时间槽周期, 架构内部时间槽切换的周期值 配置的值范围: 1-1024 个	0
submit_slot_period	31:12	R/W	保留	0
	10:0	R/W	提交时间槽周期, 架构内部时间槽切换的周期值 配置的值范围: 1-1024 个	0
report_period	31:12	R/W	保留	0
	11:0	R/W	上报周期, 配置与状态管理模块上报的周期值 配置的值范围: 1(ms) 或 1000 (ms)	0
offset_period	31:24	R/W	保留	0
	23:0	R/W	offset 补偿的配置周期	
rc_regulation_value	31:9	R/W	保留	0
	8:0	R/W	RC 流的监管阈值, 当 BUFID 的剩余个数小于该值, 开始丢弃 RC 报文	
be_regulation_value	31:9	R/W	保留	0
	8:0	R/W	BE 流的监管阈值, 当 BUFID 的剩余个数小于该值, 开始丢弃 BE 报文和 RC 报文	
unmap_regulation_value	31:9	R/W	保留	0
	8:0	R/W	非映射流的监管阈值, 当 BUFID 的剩余个数小于该值, 开始丢弃非映射报文	
reserve	31:9	R/W	保留	0

● TIS 模块

地址范围为 Addr 0x100000-0x1fffff。

表附 2-4 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
send_table_N	ST 报文发送时刻表每项内容, N=0、1、...、1023			
0x100000-0x1003ff	send_table_0 表示第 0 个发送表			

0x100400-0x1ffffff	保留
--------------------	----

表附 2-5 寄存器的具体含义

name	bit	R/W	description	default
send_table_0	16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	ST 流在一个应用周期内的注入时间槽	0
	4:0	R/W	TSNTag 中的 “send addr”	0
.....				
send_table_1023	16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	ST 流在一个应用周期内的注入时间槽	0
	4:0	R/W	TSNTag 中的 “send addr”	0
0x100400-0x1ffffff			保留	

● TSS 模块

地址范围为 Addr 0x200000-0x2fffff。

表附 2-6 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
submit_table_N 0x200000-0x2003ff	ST 报文提交时刻表每项内容, N=0、1、...、1023 submit_table_0 表示第 0 个提交表			
0x200400-0x2fffff	保留			

表附 2-7 寄存器的具体含义

name	bit	R/W	description	default
submit_table_0	31:16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	ST 流的提交时间槽	0
	4:0	R/W	TSNTag 中的 “send addr”	0
.....				

submit_table_1023	31:16	R/W	保留	0
	15	R/W	表项有效位, 0 代表无效, 1 代表有效	0
	14:5	R/W	当前 Slot	0
	4:0	R/W	TSNTag 中的 “send addr”	0

● QGC 模块

地址范围为 Addr 0x300000-0xafffff, 其中 0x300000-0x3ffff 表示第一个端口的门控表, 以此类推, 共有 8 个端口门控。

表附 2-8 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
port0_gate_table_N 0x300000-0x3003ff	0 号端口的门控表, N=0、1、...、1023, 输出门控 port0_gate_table_0 表示 0 号端口的第一个时刻的门控状态			
port1_gate_table_N 0x400000-0x4003ff	1 号端口的门控表, N=0、1、...、1023			
port2_gate_table_N 0x500000-0x5003ff	2 号端口的门控表, N=0、1、...、1023			
port3_gate_table_N 0x600000-0x6003ff	3 号端口的门控表, N=0、1、...、1023			
port4_gate_table_N 0x700000-0x7003ff	4 号端口的门控表, N=0、1、...、1023			
port5_gate_table_N 0x800000-0x8003ff	5 号端口的门控表, N=0、1、...、1023			
port6_gate_table_N 0x900000-0x9003ff	6 号端口的门控表, N=0、1、...、1023			
port7_gate_table_N 0xa00000-0xa003ff	7 号端口的门控表, N=0、1、...、1023			

表附 2-9 寄存器的具体含义

name	bit	R/W	description	default
port0_gate_table_0	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0
.....				
port7_gate_table_1	31:8	R/W	保留	0

023	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态，0 代表该队列的门控关闭，1 代表开启	0
-----	-----	-----	--	---

● FLT 模块

地址范围为 Addr 0xc00000-0xcfffff。

表附 2-10 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
0xc00000-0xc03fff	forward_table_N，表示转发表，N=0, 1, 2, ...16384, forward_table_0 表示第 0 个转发表			
0xc04000-0xcfffff	保留			

表附 2-11 寄存器的具体含义

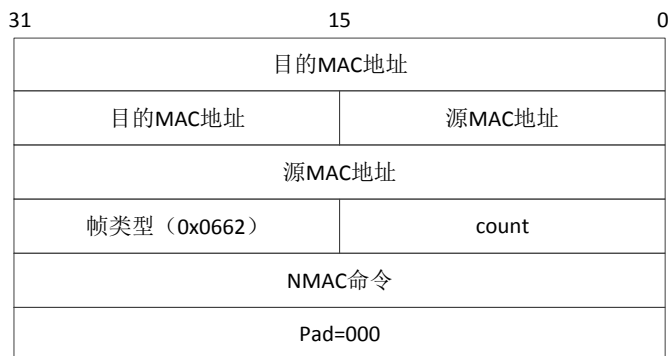
name	bit	R/W	description	default
forward_table_0	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值 0 代表不向该端口转发，1 代表向该端口转发	0
.....				
forward_table_16384	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值为 0 代表不向该端口转发，1 代表向该端口转发	0
0xc04000-0xcfffff			保留	

附录 3: NMAC 报文格式

● 配置报文格式

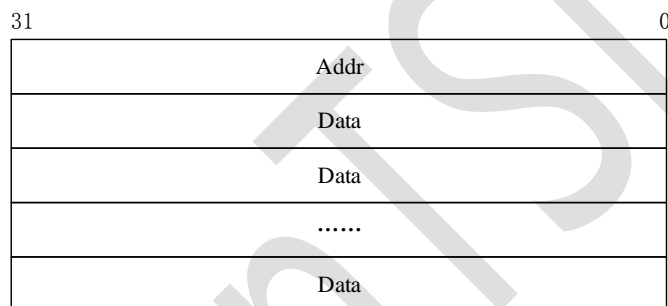
配置报文格式：在报文中用 count 字段（8bit）表示报文中包含的配置条目数，报文最小为 64 字节，最后不够 64 字节的报文需要补

零。NMAC 命令在以太网报文中的封装如图附 3-1 所示。



图附 3-1 NMAC 配置报文格式

NMAC 命令的格式如图附 3-2 所示。

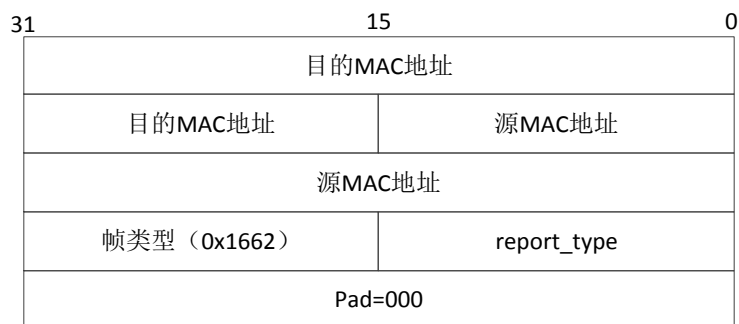


图附 3-2 NMAC 命令格式

当配置的寄存器数量为 1，NMAC 命令就包括 32bit 的 ADDR 和 32bit 的 DATA；当配置的寄存器数量为 N (N>1)，NMAC 命令就包括 32bit ADDR 和 N*32bit 的 DATA，第一个 DATA 以 ADDR 作为 RAM 写地址，第二个以及后续 DATA 以 ADDR 循环加 1 作为 RAM 写地址。

- 上报报文格式

上报报文格式：其中报文类型为 NMAC 报文 (0x1662)。



图附 3-3 NMAC 上报报文格式

表附 3-1 上报类型格式

上报类型 (16bit)		含义
高 6bit	低 10bit	
000000 单个寄存器	0	配置的单个寄存器，包含配置完成寄存器、端口状态寄存器、时间槽大小寄存器、时间偏移寄存器、上报周期寄存器、上报类型寄存器、应用周期寄存器
000001 转发表	0	第 0-63 条转发表
	1	第 64-127 条转发表
	2-255	第 128-16383 条转发表
000010 注入时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000011 提交时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000100-001011 P0-P7 输出 门控表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
001100 xx_state	0	具体参考附录 D

单个寄存器，report_type 高 6bit 为 0000，低 10bit 为 0。

转发表上报报文格式，report_type 高 6bit 为 000001，低 10bit 为上报的第几块，转发表一共有 16K 条，每条转发表占用 2 字节(9bit)，因此每个报文可以携带 64 条，总共需要 256 个报文。

注入时刻表上报报文格式, report_type 高 6bit 为 000010, 低 10bit 为上报的第几块, 注入时刻表一共有 1024 条, 每条转发表占用 2 字节 (9bit), 因此每个报文可以携带 64 条, 总共需要 16 个报文。

提交时刻表上报报文格式, report_type 高 6bit 为 000011, 低 10bit 为上报的第几块, 注入时刻表一共有 1024 条, 每条转发表占用 2 字节 (9bit), 因此每个报文可以携带 64 条, 总共需要 16 个报文。

门控表按照端口划分, 每个门控占用一块 RAM, 共有 8 个端口。每个端口两块 RAM, 总共需要 16 块 RAM。

附录 4: command/command_ack 命令格式

表附 4-1 command/command_ack 命令格式

位置	位宽	名称	说明
[203:180]	8	node_id	该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。
[179:172]	8	dest_module_id	该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID
[171:168]	4	type	4' b0001:寄存器或表项的写命令; 4' b0010:寄存器或表项的读命令; 4' b0110:寄存器或表项的读响应。
[167:152]	16	addr	寄存器或表项的读/写地址
[151:0]	152	data	寄存器或表项的读/写数据; 其中五元组映射表的表项位宽最大, 为 152bit